

Intersection and Closest-pair  
Problems for a Set of Planar Objects

by

Micha Sharir\*

Technical Report No. 56

February, 1983

CSD TR.56

C.1



Intersection and Closest-pair  
Problems for a Set of Planar Objects

by

Micha Sharir\*

Technical Report No. 56

February, 1983

\* School of Mathematical Sciences, Tel Aviv University, and  
Department of Computer Science, Courant Institute of  
Mathematical Sciences.

This research has been supported in part by ONR Grant N00014-82-K-0381  
and by a grant from the U.S. - Israeli Binational Science Foundation.



# Intersection and Closest-pair Problems for a Set of Planar Objects <sup>(\*)</sup>

Micha Sharir

School of Mathematical Sciences,

Tel Aviv University

and

Department of Computer Science,

Courant Institute of Mathematical Sciences

## ABSTRACT

Various efficient algorithms for detecting intersections and computing closest neighbors in a set of convex bodies, are presented and analyzed. These algorithms generalize known techniques for solving these problems for sets of points or of line segments. Of particular interest is the construction of a generalized Voronoi diagram for a set of  $n$  circular bodies in time  $O(n \log^2 n)$ , and its applications.

## 1. Introduction

Let  $S$  be a set of  $n$  closed convex two-dimensional bodies of relatively simple structure (e.g. circular discs, straight segments, polygons of few sides, or expansions by some amount of such objects). In this paper we consider a variety of problems associated with such a set  $S$ . Typical such problems are:

I. Do any two objects in  $S$  intersect?

II. More generally, suppose that we assign a 'color' to each object in  $S$ . Does there exist a pair of objects in  $S$  having different colors and intersecting each other?

III. If no two objects in  $S$  intersect, what is the smallest distance between any two objects in  $S$  (or, in the 'colored' version, what is the smallest distance between two objects in  $S$  having different colors)? More generally, for each object  $B$  in  $S$  find the object in  $S$  (of different color) nearest to  $B$ .

IV. Preprocess  $S$  so that, given an arbitrary 'query point'  $X$ , the object in  $S$  nearest to  $X$  can be found quickly.

Problems of this sort arise in robotics and are related to the problem of detecting and avoiding collisions between a moving subpart of a robot system and stationary objects, or between two or more moving subparts of such a system. In this note we will simplify the problem by assuming that each of the robot subparts and the stationary obstacles is either a closed convex object of a simple form, or else is covered by finitely many such objects. Note that in the 'colored' setting some of the objects involved in our problem may be known a priori to intersect (e.g. circles covering two robot subparts hinged together, or two objects covering some robot subpart which overlap

<sup>(\*)</sup> This research has been supported in part by ONR Grant N00014-82-K-0381 and by a grant from the U.S.-Israeli Binational Science Foundation.

each other may always intersect). The 'colored' version of our problems allows for this situation by looking only for intersection of subparts which would not intersect under normal conditions (e.g. subparts belonging to two distinct robot 'arms', or a robot subpart and an obstacle, etc.).

Efficient solution of these problems in the three-dimensional case would facilitate construction of an 'off-line' debugging system for robot control programs to check whether collision occurs along a planned path of motion, and would also make it more feasible to check in real time whether a moving subpart of the system is getting dangerously close to another (moving or stationary) object. The study of the two-dimensional case carried out in this paper is intended to suggest ideas which may be useful in attacking the three-dimensional case.

## 2. Detecting Intersection of Convex Objects

In this section we consider the first problem posed above and show that it can be solved in time  $O(n \log n)$  by a straightforward modification of an algorithm due to Shamos [Sh] which tests for intersection of straight line segments. We assume that the bodies in  $S$  are convex and have simple enough structure so that each of the following operations takes constant time:

- (i) Check whether two specific bodies in  $S$  intersect each other.
- (ii) For each  $B \in S$ , find the smallest and largest abscissae of points in  $B$ .
- (iii) For each  $B \in S$ , find, for a given abscissa  $x_0$ , a point  $(\bar{x}, y) \in B$ .

Let  $B_1, \dots, B_n$  be the objects in  $S$ . For each  $j=1, \dots, n$  let  $a_j$  (resp.  $b_j$ ) be the smallest (resp. largest) abscissa of a point in  $B_j$ . For simplicity we assume that the  $2n$  numbers  $a_j, b_j, j=1, \dots, n$ , are all distinct (see Fig. 1).

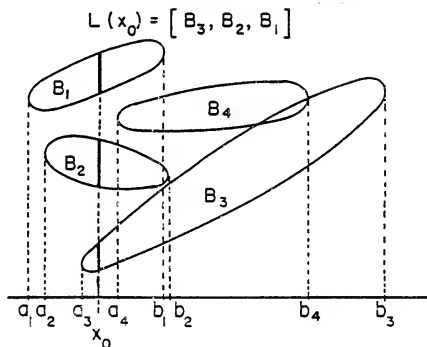


Fig. 1. An instance of the intersection problem.

The main idea used by the algorithm described below is that if we draw a vertical line  $x=x_0$  it will cut (some of) the objects in  $S$  in straight segments which, if the objects do not intersect each other along that line, are disjoint from each other. Hence the objects in  $S$  which intersect the line  $x=x_0$  can be linearly ordered in a list  $L(x_0)$  in which an object  $B_i$  precedes another  $B_j$  if the segment cut off  $B_i$  by  $x=x_0$  lies below the corresponding segment cut off  $B_j$ . Note that since the objects in  $S$  are convex, the list  $L(x_0)$  remains unchanged as  $x_0$  increases until either the

line  $z = z_0$  meets a new object  $B_k$  (this will happen when  $z_0 = a_k$ ), or when it stops making contact with some object  $B_k$  (just after  $z_0 = b_k$ ), or when two of the objects in  $L(z_0)$  intersect at  $z = z_0$ . Moreover, the leftmost intersection (if any exists) of any two objects  $B, B' \in S$  will occur at some  $z = z_0$  such that, for  $z$  slightly less than  $z_0$ , the list  $L(z)$  contains  $B$  and  $B'$  as adjacent elements. (Here we ignore the special case in which  $B$  and  $B'$  intersect at a point which is leftmost in one of these objects, which requires a slightly different argument.)

In view of these observations, to detect the presence or absence of an intersection one simply has to check repeatedly whether any two adjacent elements in the list  $L(z_0)$  intersect each other. Plainly, these checks have to be performed only at points where  $L(z_0)$  changes (i.e. at the points  $z_0 = a_j, b_j, j = 1, \dots, n$ ), and one only needs to test newly adjacent pairs in  $L(z_0)$ . To facilitate execution of these operations, the list  $L$  can be maintained as a 2-3 tree, allowing all the required list-maintenance operations to be performed in time  $O(\log n)$  (see below for details).

Details are as follows. The algorithm begins by sorting the  $2n$  numbers  $a_j, b_j, j = 1, \dots, n$ , in increasing order, and then processes them from left to right. Initially, the list  $L$  is empty. Suppose that the abscissa currently being processed is one of the  $a_j$ . Then  $L$  is updated by inserting the object  $B_j$  into  $L$  in its proper place, using a standard 2-3 tree search during which comparison of two objects  $B, B'$  is accomplished by comparing two representative points in the intersection of  $z = a_j$  with  $B, B'$  respectively (we have assumed that this can be done in constant time). After insertion, the algorithm finds the two objects  $B, B'$  immediately preceding and succeeding  $B_j$  in  $L$ , and checks whether either  $B$  or  $B'$  intersects  $B_j$ .

Similarly, if the abscissa currently being processed is  $b_j$ , then the object  $B_j$  is deleted from  $L$ , using essentially the technique just outlined. After deletion, the algorithm finds the two objects in  $L$  which immediately preceded and followed  $B_j$  prior to its deletion (these will have become newly adjacent in  $L$  after deletion of  $B_j$ ), and determines whether they intersect each other.

The algorithm halts whenever an intersection is detected, or, if no intersection has been detected, when all the abscissae  $a_j$  and  $b_j$  have been processed, in which case the algorithm reports that there is no intersection between the objects in  $S$ .

The correctness of the algorithm follows from the preceding observations. The time complexity of the algorithm is  $O(n \log n)$  since processing of each of the  $2n$  abscissae  $a_j, b_j$  can be accomplished in  $O(\log n)$  time, using a 2-3 tree representation for the list  $L$ .

### 3. Voronoi Diagrams for Circular Objects

The algorithm presented in the preceding section does not solve the more complicated 'colored object' intersection problem posed in the introduction. Indeed, the argument justifying the correctness of the algorithm breaks down as soon as an intersection is detected, so that if the first intersection detected is between two objects having the same color we can no longer use the procedure described to find additional intersection points. To handle the colored intersections problem, we therefore propose a different approach based on generalized Voronoi diagrams. We will show such an approach which can be used to handle the special case where all the objects in the set  $S$  are circular discs, not necessarily of equal radii.

Let each of the objects  $B_j \in S$  be a disc of radius  $r_j$  about the center  $z_j$ , for  $j = 1, \dots, n$ . These circular discs need not be disjoint from each other, and may intersect, or even contain, one another. We define a generalized Voronoi diagram  $Vor_0(S)$  associated with the set  $S$  as follows. For each  $i \neq j$  define

$$H(i, j) = \{y \in E^2 : d(z_i, y) - r_i \leq d(z_j, y) - r_j\},$$

i.e. the set of all points whose distance from  $B_i$  is no greater than their distance from  $B_j$  (note that the distance of  $y$  from  $B_i$  is taken as the distance of  $y$  from the boundary of  $B_i$  with a

positive sign if  $y$  lies outside  $B_i$  and with a negative sign if  $y$  lies inside  $B_i$ ). Then define the (closed) Voronoi cell  $V(i)$  associated with  $B_i$  to be

$$V(i) = \bigcap_{j \neq i} H(i, j),$$

i.e. the set of all points  $y$  whose distance from  $B_i$  is no greater than  $y$ 's distance from any other element of  $S$ . (We will sometimes refer to the point  $z_i$  as the center of this Voronoi cell; hence the center of  $V(i)$  is the same as the center of the disc  $B_i$ .) Finally, the Voronoi diagram  $Vor_0(S)$  is defined to be the set of points which belong to more than one Voronoi cell. For simplicity we assume that no point in  $Vor_0(S)$  lies in more than four Voronoi cells. This assumption generalizes the familiar assumption concerning Voronoi diagrams associated with a set of points, in which one requires that no more than four of these points be cocircular. Fig. 2. shows an example of such a Voronoi diagram.

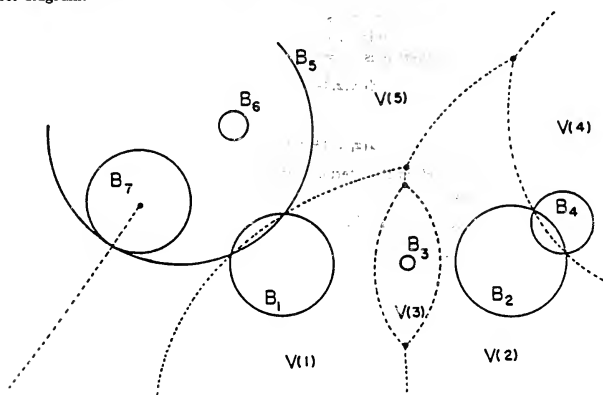


Fig. 2. The Voronoi diagram of a set of circular discs.

Generalized Voronoi diagrams have been previously introduced and analyzed by various authors; see e.g. Kirkpatrick [Ki], who considers Voronoi diagrams for a set of line segments and points, and Lee and Drysdale [LD], who consider Voronoi diagrams for sets of line segments and for sets of circular objects. Our setup is somewhat different from that of [LD], in that we allow the objects in  $S$  to intersect each other, or even to contain one another, whereas in [LD] the circles are required to be disjoint. Lee and Drysdale present an  $O(n \log^2 n)$  algorithm for constructing the Voronoi diagram of  $n$  disjoint circles, but their algorithm is given in a very sketchy form without any analysis of the structure of the diagram, and without any proof of its correctness. In addition, they assume availability of an  $O(n \log n)$  algorithm for locating  $n$  points in general (nonlinear) planar subdivisions, but give no details concerning such an algorithm (the algorithm of Preparata [Pr] to which they refer has to be modified to handle nonlinear arcs, and is unnecessarily complicated for the task of simultaneous location of  $n$  points). A by-product of the algorithm that we present in Section 4, is such a relatively simple algorithm, which may also be of independent interest.

The generalized Voronoi diagram just defined has the following properties.



(1) The collection of Voronoi cells covers the whole plane.

**Proof:** Immediate. Indeed, given any  $y \in E^2$ , it will belong to the cell  $V(i)$  for which

$$d(x_i, y) - r_i = \min\{d(x_j, y) - r_j : j=1, \dots, n\}.$$

(2)  $V(i)$  is empty iff  $B_i$  is wholly contained in the interior of another disc  $B_j$ ;  $V(i)$  has an empty interior iff  $B_i$  is wholly contained in another disc  $B_j$ .

**Proof:** For the first assertion, note that if  $V(i)$  is not empty it contains a point  $x$  such that

$$d(x, x_i) - r_i \leq \min\{d(x, x_j) - r_j : j=1, \dots, n\}.$$

The triangle inequality then shows that this same inequality holds for  $x = x_i$ , i.e.

$$-r_i = \min\{d(x_i, x_j) - r_j : j=1, \dots, n\},$$

which implies that, for any of the other circles  $B_j$  of  $S$ , the point of  $B_i$  at maximum distance from  $x_j$  is not interior to  $B_j$ . Thus there is no  $B_j$  whose interior contains  $B_i$ . Conversely, if  $V(i)$  is empty, then  $x_i \notin V(i)$ , so that there exists  $j \neq i$  such that

$$d(x_i, x_j) - r_j < -r_i,$$

i.e.

$$d(x_i, x_j) + r_i < r_j,$$

which is to say that  $B_j$  contains  $B_i$  in its interior. Similarly, if the interior of  $V(i)$  is not empty, it contains a point  $x$  such that

$$d(x, x_i) - r_i < \min\{d(x, x_j) - r_j : j=1, \dots, n\},$$

and again this inequality must hold for  $x = x_i$ . Thus

$$-r_i < d(x_i, x_j) - r_j, \quad \text{for each } j \neq i,$$

Hence  $r_j < d(x_i, x_j) + r_i$  for each  $j$ , i.e. the point of  $B_i$  at maximum distance from  $x_j$  does not belong to  $B_j$ . Thus there is no  $B_j$  which contains  $B_i$ .

Conversely, if the interior of  $V(i)$  is empty,  $x_i$  does not belong to this interior, so that there exists  $j \neq i$  such that

$$d(x_i, x_j) - r_j \leq -r_i,$$

or

$$d(x_i, x_j) \leq r_j - r_i,$$

which is to say,  $B_i$  is wholly contained in  $B_j$ .

(3)  $Vor_0(S)$  consists of straight or hyperbolic arcs.

**Proof:** Let  $y \in Vor_0(S)$  be such that  $y$  lies in both Voronoi cells  $V(i)$  and  $V(j)$ . Then we have

$$d(x_i, y) - d(x_j, y) = r_i - r_j.$$

The locus of points satisfying this condition is a hyperbolic arc having  $x_i$  and  $x_j$  as foci, or, if  $r_i = r_j$ , the perpendicular bisector to the segment  $[x_i, x_j]$ . (Note also that this (generally hyperbolic) locus degenerates into a half-line if  $d(x_i, x_j) = \pm(r_i - r_j)$ , i.e. if one of the discs  $B_i, B_j$  wholly contains the other.

(4) Each nonempty Voronoi cell  $V(i)$  is star-shaped with respect to the point  $x_i$ . Moreover, if  $V(i)$  has nonempty interior, then the interior of a segment connecting  $x_i$  to a point on the boundary of  $V(i)$  does not intersect the interior of any other Voronoi cell, and such a segment can intersect another Voronoi cell  $V(j)$  only if the corresponding disc  $B_j$  is wholly contained in  $B_i$  (so that, by

remark (3),  $V(j)$  has empty interior).

**Proof:** Let  $y \in V(i)$ , and let  $I$  be the segment connecting  $x_i$  to  $y$ . We first claim that each point  $z$  in the interior of  $I$  is contained in  $V(i)$ . Indeed, if this were false then there would exist a point  $z$  in the interior of  $I$  which does not belong to  $V(i)$ . By (1) there would exist  $j \neq i$  such that  $z$  belongs to  $V(j)$ . Hence

$$d(x_i, z) - r_i < d(x_j, z) - r_j.$$

By the triangle inequality (and since  $z$  lies between  $x_i$  and  $y$  on a straight line) we have

$$d(x_i, y) - r_i \leq d(x_i, z) + d(z, y) - r_i < d(x_i, z) + d(z, y) - r_j = d(x_i, y) - r_j.$$

Thus  $y$  cannot belong to  $V(j)$ , a contradiction which proves that  $V(i)$  is star-shaped.

Next suppose that  $V(i)$  has nonempty interior. Let  $y \in V(i)$  and  $I$  be as above, and suppose that  $I$  contains an interior point  $z$  which also belongs to some other Voronoi cell  $V(j)$ . Then we have

$$d(x_i, z) - r_i = d(x_j, z) - r_j.$$

Using the triangle inequality as before, we obtain

$$\begin{aligned} d(x_i, y) - r_i &\leq d(x_i, z) + d(z, y) - r_i = d(x_j, z) + d(z, y) - r_j \\ &= d(x_j, y) - r_j \leq d(x_i, y) - r_i, \end{aligned}$$

since  $y \in V(i)$ . Hence  $x_j$  lies on the line containing  $I$ , and  $z$  lies between  $x_j$  and  $y$  on this line. However, this implies that

$$d(x_i, x_j) = \pm (r_i - r_j),$$

i.e. that one of the discs  $B_i, B_j$  contains wholly the other. But by (2)  $B_i$  is not wholly contained in any other disc, so that  $B_j$  is wholly contained in  $B_i$ , and consequently  $V(j)$  has empty interior. This establishes our two final assertions.

(5) The intersection  $I$  of three Voronoi cells  $V(i)$ ,  $V(j)$ ,  $V(k)$  can consist of at most finitely many points; moreover, if all these three cells have nonempty interiors, then  $I$  consists of at most two points.

**Proof:** We prove only the second assertion, which is needed below, since the first assertion is much simpler. Let  $y$  be a point in  $I$ . By (4) the interior of the segment  $L_i$  (resp.  $L_j, L_k$ ) connecting  $y$  with  $x_i$  (resp.  $x_j, x_k$ ) is contained in  $V(i)$  (resp.  $V(j), V(k)$ ) and in no other cell. Let  $y'$  be another point in  $I$ , and let  $L_i', L_j', L_k'$  be the segments connecting  $y'$  with  $x_i, x_j, x_k$  respectively. It is clear that no two of these six segments can intersect one another (except at an endpoint). It follows that  $I$  cannot contain a third point  $z$ , because at least one of the segments connecting  $z$  to the three centers  $x_i, x_j, x_k$  would have to intersect one of the preceding six segments, which is impossible.

**Definition:** The modified Voronoi diagram  $Vor(S)$  is defined to consist of the boundaries of all cells in  $Vor_0(S)$  having nonempty interior. All other cells are discarded from the modified diagram.

(6) Let  $D^*$  denote the dual of  $Vor(S)$ , defined to be a graph whose vertices are the points  $x_i, i=1, \dots, n$  for which  $V(i)$  has nonempty interior, and which contains an edge  $[x_i, x_j]$  if  $V(i)$  and  $V(j)$  have a nonempty intersection including at least one open arc. If  $V(i)$  and  $V(j)$  intersect in more than one one arc, define the graph  $D^*$  to contain multiple edges connecting  $x_i$  to  $x_j$ , one for each such arc. Then  $D^*$  is a planar graph, each of whose faces (except for the outer one) contains at least three edges.

**Proof:** We define an embedding of  $D^*$  in  $E^2$  as follows. Each vertex  $x_i$  is mapped to itself (as a point in  $E^2$ ). Let  $e=[x_i, x_j] \in D^*$  be an edge corresponding to an open arc  $\alpha$  in the intersection

of  $V(i)$  and  $V(j)$ . It follows from (5) that there must exist at least one point  $y \in \alpha$  which does not belong to any other Voronoi cell. We then map the edge  $e$  to the path consisting of the two segments  $[x, y]$  and  $[y, z_j]$ . To see that the resulting graph  $G$  is indeed a planar embedding of  $D^*$ , suppose to the contrary that two distinct edges  $e_1 = [z_i, z_j]$ ,  $e_2 = [z_k, z_l]$  of  $D^*$  map to paths which intersect each other at a point  $z$  which is not a vertex of  $G$ . Let  $z_1$  (resp.  $z_2$ ) be a point on the common boundary of  $V(i)$  and  $V(j)$  (resp.  $V(k)$  and  $V(l)$ ) such that  $e_1$  (resp.  $e_2$ ) appears in  $G$  as the union of the segments  $[z_i, z_1]$  and  $[z_1, z_j]$  (resp.  $[z_k, z_2]$  and  $[z_2, z_l]$ ). Suppose without loss of generality that the segments  $s = [z_i, z_1]$  and  $s' = [z_k, z_2]$  intersect at a point other than a common endpoint  $z_i = z_k$ . Since by (4) the interior of the segment  $s$  (resp.  $s'$ ) is contained in  $V(i)$  (resp.  $V(k)$ ) and in no other Voronoi cell having nonempty interior, we must have either  $i = k$  or  $z_1 = z_2$ . In the first case  $s$  and  $s'$  meet at  $z_i$ , and so if they meet at another point they must overlap each other, which is possible only if either  $z_1 = z_2$  or if one of these points (say  $z_1$ ) lies in the interior of the other segment  $s'$ . The latter assumption would contradict the fact that the interior of  $s'$  is wholly contained in  $V(i)$  and in no other cell having nonempty interior. Thus we must have  $z_1 = z_2$ , and then by the choice of these points it follows that  $j = l$  too. Then it is plain that the two arcs of the common boundary of  $V(i)$  and  $V(j)$  which define our two paths are identical. Hence the two edges  $e_1$  and  $e_2$  are not distinct, contrary to assumption. All this shows that  $G$  is a planar embedding of  $D^*$ .

Next let  $Q$  be a face of  $G$  bounded by just two edges. Then these edges must connect the same two points  $z_i$  and  $z_j$ , through two points  $z_1, z_2$  each lying on a different open arc  $\alpha_1, \alpha_2$  of the common boundary of  $V(i)$  and  $V(j)$ . But then  $Q$  is a quadrangle whose vertices are  $z_i, z_1, z_2$ , and  $z_j$ , and must contain the point  $v$  at which one of the arcs  $\alpha_1, \alpha_2$  ends, which must be a Voronoi vertex. Clearly  $v$  must belong to the three cells  $V(i)$ ,  $V(j)$  and  $V(k)$ , where  $B_k$  is some other disc than  $B_i$  and  $B_j$ . It follows that  $z_k$  also lies inside  $Q$  (because the interior of the segment connecting  $z_k$  to  $v$  lies wholly in the interior of  $V(k)$ , and because the sides of  $Q$  are wholly contained in the union of  $V(i)$  and  $V(j)$ ). Moreover, since we have assumed that no Voronoi vertex belongs to more than three cells, it follows that  $v$  is a vertex at which three Voronoi edges meet, one of which separates  $V(i)$  from  $V(k)$  and another separates  $V(k)$  from  $V(j)$ . Hence  $Q$  contains two paths, which are images of edges  $[z_i, z_k]$  and  $[z_k, z_j]$ , and which are also contained in  $Q$ . This implies that  $Q$  is not a face of  $G$ , contrary to assumption, establishing the second assertion made above.

(7) Since  $D^*$  contains  $O(n)$  vertices, and since each of its faces contains at least three edges, it follows by Euler's formula that it has at most  $O(n)$  edges, that is,  $\text{Vor}(S)$  consists of at most  $O(n)$  connected straight or hyperbolic arcs.

(8) Let  $C$  be the convex hull of the union of all the discs  $B_i \in S$ . Note that the boundary of  $C$  consists of an alternating sequence of straight segments and circular arcs, the circular arcs being boundary portions of some of the discs, whereas the straight segments are tangents to a pair of discs in  $S$ . We will say that  $B_i$  and  $B_j$  are adjacent along the boundary of  $C$ , if this boundary contains a straight segment tangent to both  $B_i$  and  $B_j$ . Then the unbounded edges of  $\text{Vor}(S)$  are those edges that are common to two cells  $V(i), V(j)$  for which  $B_i$  and  $B_j$  are adjacent along the boundary of  $C$ .

**Proof:** Let  $e$  be an unbounded edge of  $C$ , common to two Voronoi cells  $V(i)$  and  $V(j)$ . Since  $e$  is either a straight or hyperbolic arc, it tends asymptotically to some half-line  $l$ . Suppose, without loss of generality, that  $l$  is the positive  $y$ -axis, and let  $a = [0, t]$  be a point on  $l$ . For sufficiently large  $t$  and for any  $k = 1, \dots, n$ ,  $d(a, B_k)$  behaves asymptotically as  $t - \eta_k - r_k$ , where  $\eta_k$  is the  $y$ -coordinate of  $z_k$ , i.e., as  $d(a, B_{\mu})$  where  $B_{\mu}$  is the image of  $B_k$  translated parallel to the  $x$ -axis until its center lies on the  $y$ -axis. It follows that

$$\eta_i + r_i = \eta_j + r_j \geq \eta_k + r_k,$$

for every  $k = 1, \dots, n$ . This however is easily seen to imply that  $B_i$  and  $B_j$  are adjacent to each

other along  $C$ , since the line  $y = \eta_i + r_i$  is tangent to both discs  $B_i, B_j$  and all the discs in  $S$  lie in the lower half-plane which this line bounds, so that the portion of this line between its points of tangency with  $B_i$  and  $B_j$  belongs to the boundary of  $C$ . (An extreme case that we need consider is that in which the line  $y = \eta_i + r_i$  is also tangent to a third disc  $B_k$ , at a point lying between its points of contact with  $B_i$  and  $B_j$ . However, if such a situation arises, it is easy to check that all the points on  $e$  sufficiently far away are nearer to  $B_k$  than to one of  $B_i, B_j$ , contradicting the definition of  $e$ . The converse statement, namely that any pair of adjacent discs along  $C$  induces an unbounded Voronoi edge, can be proved using the above argument in reverse.

(9)  $Vor(S)$  need not be connected. In fact, it can have up to  $O(n)$  connected components. However, every connected component of  $Vor(S)$  is unbounded.

**Proof:** Consider the following set  $S$  of discs, which consists of the unit disc  $B_1$ , and of  $k$  additional small discs  $B_2, \dots, B_{k+1}$  all of radius  $\rho$ , such the centers of these discs are placed on the boundary of  $B_1$  at equally spaced positions. If  $\rho$  is chosen to be sufficiently small (e.g. of the order  $O(\frac{1}{k^2})$ ) then it is easily checked that for each  $j=2, \dots, k+1$  the discs  $B_1$  and  $B_j$  are adjacent to each other along the boundary of the convex hull of the  $B_i$ 's. Moreover, it can also be shown that for  $\rho$  sufficiently small each unbounded edge common to  $V(1)$  and some other  $V(j)$  is a full branch of the corresponding hyperbola, and that no two such edges intersect each other.

This shows that  $Vor(S)$  can have as many as  $O(n)$  connected components.

**Remark:** The example just given also shows that the boundary of a single Voronoi cell ( $V(1)$  in the example) can have up to  $O(n)$  disjoint connected components.

Suppose next that for some set  $S$  of discs  $Vor(S)$  contains a bounded component  $K$ . Then the portion  $E$  of a sufficiently small neighborhood of  $K$  which lies exterior to  $K$  must be contained in a single Voronoi cell  $V(i)$ , since otherwise some arc of  $Vor(S)$  would have to enter any such exterior neighborhood of  $K$ , contradicting the assumption that  $K$  is a connected component of  $Vor(S)$ . But if a whole neighborhood of  $K$  in the exterior of  $K$  lies in  $V(i)$ , there must exist a point  $y \in \text{int}(V(i))$  such that the line connecting  $y$  to  $z_i$  intersects  $K$ , contradicting (6). Thus  $Vor(S)$  cannot have any bounded connected component.

**Corollary:**  $Vor(S)$  does not contain any isolated point. Moreover, by modifying the argument given above one can also show that each Voronoi vertex must belong to at least two edges. (Assertion (11) below will strengthen this claim, by showing that each such vertex must belong to three distinct Voronoi edges.)

(10) No two edges of  $Vor(S)$  can be tangent to each other. Also, for each point  $z$  on a Voronoi edge  $e$  separating two cells  $V(i)$  and  $V(j)$ , the segment connecting  $z$  to  $z_i$  (or to  $z_j$ ) is not tangent to  $e$ .

**Proof:** Since Voronoi edges are either straight segments or hyperbolic arcs it follows that if two Voronoi edges are tangent to each other then any Voronoi edges lying between them at their point of tangency must be tangent to both of them. Hence if there exist any two tangent Voronoi edges, then there exist two such edges which belong to the boundary of the same Voronoi cell. Assume this to be possible, and let  $V(i)$  be a Voronoi cell whose boundary contains two edges  $e, e'$  which also belong to  $V(j), V(k)$  respectively, and which are tangent to each other at some point  $y$ . It is plainly impossible for both  $e$  and  $e'$  to be straight arcs; hence at least one must be part of a hyperbola. Let  $l$  be the line which is tangent at  $y$  to both hyperbolas (or to the hyperbola and straight line) containing  $e, e'$  respectively. As is well known, a line tangent to a hyperbola at a point  $y$  bisects the angle  $f_1 y f_2$ , where  $f_1$  and  $f_2$  are the two foci of the hyperbola; furthermore, this result also holds trivially in case the hyperbola degenerates into the perpendicular bisector of the segment connecting the two points  $f_1, f_2$ . Thus, in any case  $l$  bisects the angle between the two segments connecting  $y$  to  $z_i$  and to  $z_j$  (resp. to  $z_i$  and  $z_k$ ). It follows that the these two angles must be equal, and consequently the three points  $y, z_i, z_k$  are colinear, with  $z_i$  and  $z_k$  lying on the same side of  $y$ . Suppose for definiteness that  $z_j$  lies between  $y$  and  $z_k$ . By

definition of  $e$  and  $e'$  we then have

$$d(x_i, y) - d(x_j, y) = r_i - r_j,$$

$$d(x_i, y) - d(x_k, y) = r_i - r_k$$

or

$$d(x_k, y) - d(x_j, y) = d(x_j, x_k) = r_k - r_j,$$

which is to say,  $B_j$  is wholly contained in  $B_k$ , so that by definition  $Vor(S)$  does not contain any edge bounding  $V(j)$ . This contradiction establishes our assertion.

The second assertion follows from the fact that no tangent to a hyperbola can pass through any of its foci. (The only exception is when the hyperbolic arc containing  $e$  degenerates into a half-line, but then either  $B_i$  or  $B_j$  must be wholly contained in some other disc, so that by convention  $e$  does not appear in our Voronoi diagram.)

(11) Let  $e$  and  $e'$  be two adjacent edges along the boundary of a Voronoi cell  $V(i)$ . Then the interior angle in  $V(i)$  between  $e$  and  $e'$  is less than 180 degrees. (Stated otherwise, in traversing the boundary of  $V(i)$  to the right, we make a right turn as we pass from one of the edges  $e, e'$  to the other.

**Proof:** We have already shown that  $e$  and  $e'$  cannot be tangent to each other. Let  $y$  be their point of intersection, and let  $l$  be the line containing  $x_i$  and  $y$ . The two edges  $e$  and  $e'$  cannot both lie on the same side of  $l$  in the vicinity of  $y$ , because then the segment connecting  $x_i$  to a point on one of them sufficiently near  $y$  would have to intersect the other edge, contradicting (6). Suppose then that  $e$  lies on the left side of  $l$  (oriented from  $x_i$  to  $y$ ), and that  $e'$  lies on the right side of  $l$ . Extend  $e$  along the hyperbolic branch containing it into the right side of  $l$  (note that a hyperbola is never tangent to the segment connecting a point lying on it to one of its foci), and denote the extended portion of  $e$  by  $e''$ . Then  $e'$  must lie between  $x_i$  and  $e''$ , for otherwise the segment connecting  $x_i$  to any point on  $e'$  lying sufficiently near to  $y$  will intersect  $e''$  at a point  $z$ , and plainly no point  $z \in e''$  can belong to the interior of  $V(i)$  (because  $z$  is equidistant from  $B_i$  and from some other disc in  $S$ ), again contradicting (6). It therefore follows that the interior angle between  $e$  and  $e'$  is less than 180 degrees, as asserted.

**Corollary:** This argument shows that each Voronoi vertex must be incident to three distinct Voronoi edges, for if it belonged to just two edges, at least one of the angles between these edges would be interior to some Voronoi cell, and would be greater than 180 degrees, contrary to what we have just shown.

#### 4. Efficient Construction of Generalized Voronoi Diagrams.

Next we present an algorithm which generalizes the divide-and-conquer methods used by Shamos [Sh] and by Kirkpatrick [Ki] to construct the Voronoi diagram of a set of points, and which computes the Voronoi diagram of a set  $S$  of  $n$  circular bodies in time  $O(n \log^2 n)$ . The algorithm produces a list of all Voronoi cells having nonempty interior, and for each such cell constructs a circular list containing the edges on its boundary, arranged in clockwise order (for unbounded cells this list will also include 'virtual edges' at infinity connecting pairs of unbounded edges  $e, e'$  such that the intersection of  $e'$  with an arbitrarily large circle lies immediately clockwise to the intersection of this circle with  $e$ ). Finally, the algorithm produces a table in which each edge points to the two cells containing it.

For simplicity, we will assume in what follows that no two circles in  $S$  have distinct leftmost points lying on the same vertical line. If  $S$  does not have this property, then we can apply an infinitesimal rotation that will make the abscissae of the leftmost points of all circles in  $S$  distinct from each other. Cf. also [SS] where a similar technique based on infinitesimal perturbations is used to resolve degenerate configurations arising in other geometric problems. The algorithm

begins by dividing  $S$  into two subsets  $R$  and  $L$  of equal size, such that the leftmost point  $w_i$  of each  $B_i \in L$  lies to the left of the leftmost point  $w_j$  of every disc  $B_j \in R$ . This partitioning of  $S$  can easily be done in time  $O(n)$ . Note that it has the property that no body  $B_i \in L$  is wholly contained in another body  $B_j \in R$ , although a reverse containment is possible.

Assume that the Voronoi diagrams  $Vor(R)$  and  $Vor(L)$  have been computed recursively. The main step of the algorithm is to merge these two diagrams into a single diagram  $Vor(S)$ . For this, one must compute the set  $C$  of points  $y$  which are simultaneously nearest to a disc  $B_i \in L$  and to a disc  $B_j \in R$ . Following Kirkpatrick [Ki] We call  $C$  the contour separating  $R$  and  $L$ . We will see that, once  $C$  has been computed,  $Vor(S)$  can be obtained by taking the union of  $Vor(R)$ ,  $Vor(L)$ , and  $C$ , and then by discarding (portions of) edges belonging to one of the partial diagrams  $Vor(R)$ ,  $Vor(L)$ , whose points have become nearer to some object belonging to the other set (these portions will be delimited by the intersections of these edges with  $C$ ). Note that during this merging step some Voronoi cells  $V(i)$  with  $B_i \in R$  may be wholly deleted from  $Vor(S)$  if  $B_i$  happens to lie wholly inside some disc  $B_j \in L$ .

Since  $C$  will be a curve consisting of straight and hyperbolic arcs, the complement of  $C$  will consist of finitely many open connected planar regions. Each such region  $M$  is either a union of cells  $V(i)$  (in the final diagram  $Vor(S)$ ) with  $B_i \in L$  (in which case we will call  $M$  an L-region), or a union of cells  $V(i)$  with  $B_i \in R$  (in which case  $M$  will be called an R-region).

By assertion (9) of Section 3, the contour  $C$  can consist of several disjoint connected components. Moreover, it is also possible for a connected component of  $C$  to be bounded, as shown by the example appearing in Fig. 3.

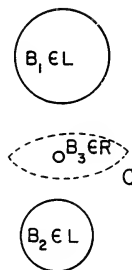


Fig. 3. A bounded connected component of  $C$ .

Fig. 3 shows that the contour  $C$  can be moderately complex; the following lemma restricts this complexity, and begins to develop some of the facts that will be needed to show that tracing all the components of  $C$  need not be expensive.

**Lemma 4.1:** (a)  $C$  consists of a disjoint union of simple topologically closed curves without end-points (i.e. each such curve is either closed or stretches to infinity in both directions).

(b) Let  $B_i \in L$  and let  $u$  be the horizontal half-line whose rightmost endpoint is  $z_i$ . Then  $u$  does not intersect  $C$  (and consequently lies wholly inside an L-region).

(c) There exists precisely one L-region; all other components of the complement of  $C$  are R-regions.

(d) Each R-region has a connected boundary, consisting of a single (bounded or unbounded) component of  $C$ .

**Proof:** (a) It follows from its definition that  $C$  is a union of Voronoi edges of  $Vor(S)$ . It therefore suffices to show that for each Voronoi vertex  $v$  of  $Vor(S)$  lying on  $C$ , there are exactly two Voronoi edges emerging from  $v$  which belong to  $C$ . Since we have ruled out degenerate configurations, we can assume that  $v$  belongs to exactly three Voronoi cells  $V(i)$ ,  $V(j)$  and  $V(k)$ . Moreover, since  $v \in C$ , one of the discs  $B_i$ ,  $B_j$ ,  $B_k$  must belong to  $L$ , and another of these discs must belong to  $R$ . Assume first that  $B_i, B_k \in L$  and that  $B_j \in R$ . Then, in the neighborhood of  $v$ , the contour  $C$  consists of the two edges separating  $V(j)$  from  $V(i)$  and  $V(k)$  respectively. Much the same argument applies if  $B_i, B_k \in R$  and  $B_j \in L$ . This proves (a).

(b) Suppose the contrary, and let  $z$  be a contour point lying on  $u$ . Let  $w \in u$  be a point at distance  $r_i$  from  $z_i$  (that is,  $w$  is the leftmost point of  $B_i$ ). It follows that there exist discs  $B_j \in R$  (which is not wholly contained in any other disc), and  $B_k \in L$  such that

$$d(z, B_j) = d(z, B_k) \leq d(z, B_i) = d(z, w)$$

But then some point on  $B_j$  (and in particular its leftmost point) must lie to the left of  $w$ , or coincide with  $w$ . However, both these possibilities contradict the way in which  $L$  and  $R$  have been defined, a contradiction which proves (b).

(c) It suffices to show that the centers  $z_i, z_j$  of each pair  $B_i, B_j$  of discs in  $L$  are connected to each other via a path which does not intersect the contour  $C$ . Let  $u_i$  (resp.  $u_j$ ) be the horizontal half-line whose rightmost endpoint is  $z_i$  (resp.  $z_j$ ), and let  $w_i \in u_i$  (resp.  $w_j \in u_j$ ) be the leftmost point of  $B_i$  (resp.  $B_j$ ). For each  $t$ , let  $y_t$  (resp.  $y_j$ ) be a point on  $u_i$  (resp.  $u_j$ ) whose abscissa is  $t$ . We claim that if  $t$  is negative and has a large enough absolute value, the segment  $e = y_i y_j$  does not intersect the contour, which, together with (b), implies that  $z_i$  and  $z_j$  are connected to each other via the polygonal path  $z_i y_i y_j z_j$  which is wholly contained in a single L-region. To see this, suppose to the contrary that there exists a contour point  $z \in e$ , nearest to some  $B_k \in R$  and to some  $B_m \in L$ . Then

$$d(z, B_k) = d(z, B_m) \leq d(z, B_i) = (d^2(y_i, w_i) + d^2(z, y_i))^{\frac{1}{2}} \approx d(y_i, w_i)$$

and similarly

$$d(z, B_k) \leq d(z, B_j) \approx d(y_j, w_j)$$

as  $t \rightarrow \infty$ . It follows that the leftmost point  $w_k$  of  $B_k$  does not lie to the right of  $w_i$  or of  $w_j$ . Since, by definition of  $L, R$ ,  $w_k$  cannot lie to the left of  $w_i, w_j$ , it follows that the three points  $w_i, w_j, w_k$  all have the same abscissa, again contradicting the way in which  $L$  and  $R$  are defined.

(d) Suppose that  $M$  is an R-region whose boundary consists of at least two disjoint connected components  $K_1, K_2$ . By part (a) of the present lemma  $K_1$  (resp.  $K_2$ ) partitions the plane into two disjoint components, both bounded by  $K_1$  (resp.  $K_2$ ). Since  $K_1$  and  $K_2$  are disjoint, it follows that they both collectively partition the plane into three components, one of which contains  $M$ , whereas the other two contain L-regions. This, however, contradicts (c), thus proving our assertion. Q.E.D.

In general outline, the remainder of our argument is as follows. We first show that, given points  $z_K$  on each of the components  $K$  of  $C$ , the whole of  $C$  can be traced in a number of steps bounded by  $O(n)$ . Next we show how to find such a set of points  $z_K$ . This is done by noting that (by definition) every R-region must contain at least one center  $z_i$  of a disc  $B_i \in R$  such that  $B_i$  is not contained in any other disc of  $R$ . Hence if we iterate over all such points  $z_i$ , and connect each one of them by a straight arc  $e$  to a point of an L-region, these  $e$  must together intersect all

the components of  $C$ . We will show that in total time  $O(n \log n)$  we can find such arcs  $e$ , each intersecting  $C$  in just one point  $z$ , and in this way can find a  $z$  on each component  $K$  of  $C$ . This leads to an  $O(n \log n)$  merging step, and hence to an  $O(n \log^2 n)$  overall algorithm.

To facilitate the tracing of  $C$  during the merge phase of our algorithm, we will find it convenient (following [Ki]) to partition each Voronoi cell  $V(i)$  (of either  $Vor(L)$  or  $Vor(R)$ ) into subcells by connecting  $z_i$  to each vertex  $v$  of  $V(i)$  by a straight segment (called a *spoke*, as in [Ki]). Clearly each subcell is an angular sector bounded by two spokes and one Voronoi edge. Note that, given a directed straight line or hyperbolic arc  $e$ , its intersection points with the boundary of any Voronoi subcell can be found in constant time, assuming that we use an appropriate representation of the corresponding diagram  $Vor(L)$  or  $Vor(R)$ .

Suppose that we have somehow found a point  $z \in C$  (but such that  $z$  is not in either  $Vor(L)$  or  $Vor(R)$ ), for which the two discs  $B_i \in L$ ,  $B_j \in R$  nearest to  $z$  are known, and suppose further that the two subcells of  $V(i)$  in  $Vor(L)$  and of  $V(j)$  in  $Vor(R)$  to which  $z$  belongs are also known. Then we can trace the component  $K$  of  $C$  containing  $z$  as follows. We first find the Voronoi edge  $e$  (in  $Vor(S)$ ) containing  $z$ . Note that  $e$  is part of the straight line or hyperbolic arc equidistant from  $B_i$  and  $B_j$ , and is an edge lying on  $K$ . We begin tracing  $K$  by following  $e$  from  $z$  in some direction, and by computing its intersection points with the boundaries of the two subcells  $U(i)$ ,  $U(j)$  of  $V(i)$ ,  $V(j)$  respectively, containing  $z$ . Suppose for specificity that the nearest of these points along  $e$  is the point  $z'$  at which  $e$  intersects the boundary of  $U(i)$ . If  $z'$  lies on a Voronoi edge, then the contour  $K$  crosses this edge to another Voronoi cell  $V(k)$  of  $Vor(L)$  after  $z'$  (by assertion (10) of the previous section, two Voronoi edges are never tangent to one another). In this case,  $K$  continues after  $z'$  along the Voronoi edge  $e'$  containing points equidistant from  $B_i$  and  $B_j$ . On the other hand, if  $z'$  lies on a spoke,  $K$  will continue after  $z'$  along the edge  $e$ , but will cross into another subcell of  $V(i)$  (Note that the contour can never be tangent to a Voronoi spoke, by the second part of assertion (10)).

Tracing the contour in this way, we either come back to  $z$ , in which case the component  $K$  is a bounded component of the contour, or else we reach an unbounded edge of the contour, in which case we have to repeat the tracing procedure just outlined by starting from  $z$  in the other direction of the edge  $e$  in order to obtain the entire component  $K$ .

Let  $M$  be the R-region bounded by  $K$ . Each cell  $V(i)$  through which  $K$  passes is cut by  $K$  into several portions, one of which belongs to an R-region (and contains  $z_i$ ) while the others belong to the L-region. Moreover, all the cell portions belonging to R-regions belong to the same R-region  $M$ . Thus, as  $K$  is being traced, we can also note that all these cell portions cut by  $K$  belong to  $M$ . Observe that  $M$  may also contain additional internal cells of  $Vor(R)$  which have not yet been encountered. These will be dealt with during later steps of the algorithm.

Next we show that if a point  $z_K$  is available on each of the components  $K$  of  $C$ , the total cost of constructing  $C$  is  $O(n)$ . The complexity of the tracing procedure just described is plainly  $O(n_1^{(K)} + n_2^{(K)})$ , where  $n_1^{(K)}$  is the number of Voronoi edges in  $K$ , and where  $n_2^{(K)}$  is the number of intersections of  $K$  with Voronoi spokes (in either  $Vor(L)$  or  $Vor(R)$ ). As in [Ki], we can show that the sum over all  $K$  of the quantities  $n_1^{(K)} + n_2^{(K)}$  is  $O(n)$ . Namely we have

**Lemma 4.2:** Each Voronoi spoke (in either  $Vor(L)$  or  $Vor(R)$ ) is intersected by the contour  $C$  in at most one point. Moreover this remark also holds for any segment one of whose endpoints is the center  $z_i$  of some Voronoi cell  $V(i)$  (in either diagram) and which is wholly contained in  $V(i)$ .

**Proof:** Let  $e$  be a Voronoi spoke of a cell  $V(i)$  in  $Vor(L)$ . Then each point  $z$  at which  $e$  and  $C$  intersect each other must lie on the boundary of the cell  $V(i)$  in  $Vor(S)$ , and since this cell is star-shaped with respect to  $z_i$ , it follows that at most one such intersection point can exist. Q.E.D.

**Corollary:** The total time required to trace the contour, given a point  $z$  (and the two subcells in  $Vor(L)$ ,  $Vor(R)$  containing it) on each of its connected components, is  $O(n)$ .



**Proof:** The total number of edges on  $C$  is  $O(n)$ , because  $C$  is a subset of  $Vor(S)$ . The total number of intersections of  $C$  with Voronoi spokes is also  $O(n)$ , by Lemma 4.2. Hence the total complexity of the tracing procedure applied to each component  $K$  of  $C$  is

$$\sum_K (n_1^{(K)} + n_2^{(K)}) = O(n)$$

Q.E.D.

The problem that now remains is that of finding a representative point  $z_K$  on each component  $K$  of the contour  $C$  (and also finding the subcells containing  $z_K$ ). For this, Kirkpatrick [Ki] uses a technique which traces edges of minimum spanning trees for  $R$  and  $L$ . However, this technique, which works nicely for a set  $S$  of points, is not easily generalizable to sets of more general objects like the circles which now concern us. We will therefore present an alternative approach, which works for sets of circular discs, but whose complexity is  $O(n \log n)$ , instead of the linear complexity of Kirkpatrick's technique.

We iterate over all the cells  $V(i)$  of  $Vor(R)$  (and their corresponding circles  $B_i$ ), proceeding as follows. Let  $B_j \in L$  be such that  $z_i \in V(j)$  (in  $Vor(L)$ ) so that  $B_j$  is the circle of  $L$  which is 'closest' to  $z_i$ . If  $B_j$  contains all of  $B_i$ , then, by definition, the final diagram  $Vor(S)$  will not contain a cell  $V(i)$ . In this case we simply do not use the pair  $z_i, z_j$  to find a point on the contour, but go on to consider the other discs  $B_{i'} \in R$ .

Next suppose that  $B_j$  does not contain all of  $B_i$ . Then no other disc  $B_k \in L$  can have this property, and so it follows that the disc  $B_j$  in  $S$  for which  $d(z_i, x_i) - r_i$  reaches its minimum belongs to  $R$ . Similarly,  $d(z_i, x_i) - r_i$  attains its minimum when  $i = j$ , since no disc in  $R$  can contain the whole of a disc in  $L$ , and since the recursive construction of  $Vor(L)$  described in the following paragraphs will eliminate discs that are wholly contained in other discs. Thus the segment  $e = z_i, z_j$  must contain a point  $z$  for which  $d(z, x_i) - r_i$  reaches its minimum simultaneously for some  $B_i \in L$  and for some other disc  $B_{i'} \in R$ , which is to say, a point  $z$  on  $C$ . Moreover, since  $e$  emanates from  $z_i$  and is wholly contained in  $V(j)$ , it follows by Lemma 4.2 that the contour  $C$  cannot intersect  $e$  in more than one point. Hence  $e$  intersects  $C$  in exactly one point. Note that the entire segment  $e$  is contained in a single subcell of  $V(j)$  in  $Vor(L)$ . All that we have to show is that we can either find  $z$ , or assure ourselves that a point on the same component  $K$  (of  $C$ ) as  $z$  has already been found, in total time  $O(n \log n)$ .

We can proceed to find the unique intersection  $z$  of  $e$  with  $C$  using a technique quite similar to the contour-tracing procedure described above. That is, we first find the Voronoi subcell (of  $V(i)$ ) in  $Vor(R)$  containing points on  $e$  near  $z_i$  (since  $e$  emerges from  $z_i$ , this amounts to finding the two Voronoi spokes of  $V(i)$  between which  $e$  lies). We then find the intersection of  $e$  with the Voronoi edge bounding that subcell, beyond which  $e$  crosses into another subcell of  $Vor(R)$ . (In the extreme case in which  $e$  coincides with a Voronoi spoke of  $V(i)$ ,  $e$  will exit the two subcells of  $V(i)$  in which it lies at the Voronoi vertex which is the other endpoint of the spoke; it is then a bit more complicated, but still straightforward, to determine the Voronoi subcell into which  $e$  enters past this vertex.) Continuing in this manner, we partition  $e$  into subsegments  $e_1, \dots, e_m$ , each of which is contained in some Voronoi subcell of  $Vor(R)$  or lies along a spoke of  $Vor(R)$ . As all this is done, we keep track of all the cells  $V(k)$  of  $Vor(R)$  that have already been encountered. If such a cell is encountered for the second time, tracing of the sequence of edges  $e_1, \dots, e_m$  stops immediately (this rule is justified by Lemma 4.3 below). This guarantees that the total cost of traversing subcells of  $Vor(R)$  is bounded by the total number of such subcells, and hence by  $O(n)$ .

For each  $i=1, \dots, m$  let  $V(i_i)$  be the cell in  $Vor(R)$  containing  $e_i$ . As tracing proceeds through the cell  $V(i_i)$ , we check whether  $V(i_i)$  has been encountered before, and, if not, whether there exists  $z \in e_i$  such that  $d(z, B_i) = d(z, B_{i'})$  (this can be done in constant time). As already shown, there will exist a unique point  $z$  on  $e$  having this property, and this  $z$  will be the required

intersection point of  $e$  with  $C$  (the algorithm will reach this  $z$  only if tracing is not abandoned earlier, because a previously encountered cell of  $Vor(R)$  is encountered again). Let  $e_s$  be the subsegment of  $e$  containing  $z$ . Note that  $z$  is found in time  $O(p+s)$ , where  $p$  is the number of subcells of  $V(i)$  in  $Vor(R)$ .

To show that tracing of  $e$  can be abandoned as soon as any cell of  $Vor(R)$  is encountered for the second time, we will use the following

**Lemma 4.3:** Let  $s$  be as in the preceding paragraph. Then for each  $t \leq s$ ,  $Vor(S)$  contains a cell whose center is  $x_t$ , and the point  $x_t$  lies in the same R-region as  $x_1$ .

**Proof:** Let  $M$  be the R-region containing  $x_{t_1} = x_1$ , and let  $1 < t \leq s$ . Pick any point  $z_t \in e_t$  (but in case  $t = s$ ,  $z_t$  must lie between  $x_t$  and  $z$ ). Since  $t \leq s$ , the segment  $x_1 z_t$  does not intersect the contour, and therefore is wholly contained in  $M$ . As always, let  $B_t$  be the disc in  $L$  corresponding to the cell  $V(i_t)$  of  $Vor(R)$ , and let  $x_t$  be its center. Suppose for the moment that we have already shown that a cell  $V(i_t)$  (with center  $x_t$ ) appears in  $Vor(S)$ , i.e. that  $B_t$  is not wholly contained in any disc of  $L$ . The segment  $J = x_1 z_t$  is contained in the cell  $V(i_t)$  of  $Vor(R)$ , which is star-shaped with respect to its center  $x_t$ , by property (4) of Section 3. Moreover, since this segment emanates from  $x_1$ , it can intersect the contour in at most one point, by Lemma 4.2. But such an intersection is impossible, because both the endpoints of  $J$  lie in an R-region ( $x_1$  lies in an R-region because by assumption it belongs to  $V(i_1)$  in  $Vor(S)$ ). Therefore the polygonal path  $x_1 z_t$  does not intersect the contour. But  $x_1$  and  $x_t$  are connected to each other via this path, and hence lie in the same R-region.

It only remains to show that a cell  $V(i_t)$  with center  $x_t$  appears in  $Vor(S)$ . For this, note that the point  $z_t$  lies in an R-region, so that it is nearer to some disc of  $R$  than to any disc of  $L$ . From this it is plain that  $z_t$  is nearer to  $B_t$  than to any other disc of  $S$ . But then  $z_t$  must be an interior point of the cell  $V(i_t)$  in  $Vor_0(S)$ , so that, by definition of  $Vor(S)$ , a cell  $V(i_t)$  with center  $x_t$  appears in this diagram, as asserted. Q.E.D.

As we apply the procedure just described to each of the discs  $B_i \in R$ , one of the following three situations will arise: either

- (a) We discard  $B_i$  immediately, because the nearest disc  $B_j$  in  $L$  wholly contains  $B_i$ ; or
- (b) While tracing subcells of  $Vor(R)$  crossed by the segment  $e = x_1 z_j$  (where  $B_j \in L$  is the disc for which  $x_j \in V(j)$ ), we encounter a subcell of some cell  $V(r)$  whose R-region  $M$  in  $Vor(S)$  was encountered before. In this case we conclude from Lemma 4.3 that  $B_i$ , as well as every other disc of  $R$  whose cell in  $Vor(R)$  has been crossed by  $e$  so far, lies in the R-region  $M$ . In this case we can stop tracing  $e$  and go on to process other discs of  $R$ , since we can be sure that the component of the contour  $C$  intersected by  $e$  has already been explored. The algorithm will also note that all cells of  $Vor(R)$  crossed by  $e$  so far belong to  $M$ , to avoid repeated processing of these cells later on. (Note that this case will arise only when  $V(i)$  is an inner cell in  $M$ , i.e. a cell not intersected by the contour); or
- (c) The tracing procedure continues till an intersection  $z$  of  $e$  with the contour is found. In this case only new cells of  $Vor(R)$  are being traced, and  $z$  will lie on a new component of  $C$ . As in (b), we take note of the fact that all cells crossed by  $e$  during this tracing belong to the new R-region just found, to avoid repeated processing of these cells later on.

These observations imply that we can find a representative point on each component of the contour in total time  $O(n)$ , provided that, for each  $B_i \in R$ , the subcell of the cell  $V(i)$  of  $Vor(L)$  containing  $x_i$  is already known.

To obtain this final item of information, we can use a simple plane-sweeping algorithm, similar to those described by [Sh], [BO], [NP]. The algorithm sweeps the plane from left to right

and maintains a vertical "front"  $T(a)$  consisting of the segments lying along the line  $x = a$  and delimited by the points of intersection of this line with the edges and spokes of  $Vor(L)$ . The structure of  $T(a)$  will change only at points  $a$  which are either abscissae of Voronoi vertices of  $Vor(L)$ , or abscissae of centers of discs in  $L$ , or points for which the line  $x = a$  is tangent to some Voronoi edge of  $Vor(L)$ . The number of such 'transition points' is plainly  $O(n)$ , and the total number of segments of  $T(a)$  is also  $O(n)$  for any real  $a$ . To start the algorithm, sort the set  $A$ , consisting of all transition points of  $Vor(L)$  and all centers  $z_i$  of discs  $B_i \in R$ , by their  $x$ -coordinates, and initialize the list  $T(a)$  as a 2-3 tree for some large enough negative real  $a$ . Both these tasks can be done in time  $O(n \log n)$ . Then scan  $A$  from left to right. For each  $a \in A$ , if  $a$  is a transition point of  $Vor(L)$ , update the list  $T$  by an appropriate combination of deletions, insertions, and merge operations applied to segments in  $T$ ; this can be done in time  $O(k_a \log n)$ , where  $k_a$  is the number of segments which undergo these changes. Note that if  $a$  is the abscissa of a center  $z_i$  of some disc  $B_i \in L$ , then  $k_a$  is the number of Voronoi edges on the boundary of the cell  $V(i)$  in  $Vor(L)$ , which may be large. Nevertheless, the total sum of all the  $k_a$ 's over all transition points  $a$  is always  $O(n)$ . If  $a$  is the abscissa of a center  $c$  of some disc in  $R$ , search  $T$  to find the segment in  $T$  containing  $z_i$ , from which the Voronoi subcell of  $Vor(L)$  containing  $c$  is readily obtained. Proceeding in this way, we locate all the centers of discs of  $R$  in  $Vor(L)$  in time  $O(n \log n)$ .

Together, the details just described yield the following algorithm for constructing  $Vor(S)$ :

1. Split  $S$  into two equal-size subsets  $L, R$  such that the leftmost point of each  $B_i \in L$  lies to the left of the leftmost point of every  $B_j \in R$  (we have assumed that no two leftmost points have the same abscissa).
2. Compute  $Vor(L)$  recursively.
3. Apply the plane-sweeping procedure described above to locate the subcell  $V(j)$  of  $Vor(L)$  containing  $z_j$  for all centers  $z_j$  of discs  $B_j \in R$ . Discard the disc  $B_j \in R$  if it is wholly contained in  $B_j$ .
4. Let  $R'$  be the remaining set of discs of  $R$ . Compute  $Vor(R')$  recursively.
5. Construct the 'contour'  $C$  as follows:

For each disc  $B_i \in R$  whose R-region (in  $Vor(S)$ ) has not yet been identified

- a. Connect  $z_i$  to the center  $z_j$  of the disc  $B_j \in L$  Whose Voronoi cell  $V(j)$  in  $Vor(L)$  contains  $z_i$ . If  $B_j$  wholly contains  $B_i$ , then  $Vor(S)$  will not contain a cell corresponding to  $B_i$ , and we go on to process other discs of  $R$ .
  - b. Find the unique intersection  $z$  of the contour with the segment  $e = z_i z_j$  by applying the tracing procedure described above. If that procedure detects an intersection of  $e$  with a cell  $V(k)$  of  $Vor(R)$  whose R-region  $M$  has already been found, it assigns  $M$  as the R-region of  $B_i$  and of all other discs of  $R$  whose cells in  $Vor(R)$  have been crossed by  $e$  before  $V(k)$  has been reached, and continues with the main loop of this phase.
  - c. Trace the whole contour component  $K$  containing  $z$ . An R-region indication is thereby assigned to all discs  $B_k \in R$  whose cells in  $Vor(R)$  are encountered.
6. Obtain the final diagram  $Vor(S)$  by taking the union of  $Vor(R)$  and of  $Vor(L)$  with  $C$ , and then by discarding (portions of) edges of  $Vor(R)$  or of  $Vor(L)$  which are cut off from their cells by  $C$ .

The algorithm just sketched runs in time  $O(n \log^2 n)$ . Its costliest phase is step 3, which locates subcells of  $Vor(L)$  containing the centers of discs of  $R$ .

Note that it is a simple matter to modify the algorithm so that it also produces a mapping *contain*, which, for each disc  $B_i$  deleted from the diagram by the algorithm, gives the disc  $B_j$  containing  $B_i$  as found in step 5.a of the algorithm.

## 5. Applications of the Generalized Voronoi Diagram; Possible Extensions.

In this section we show how the generalized Voronoi diagram can be used to solve some of the intersection problems mentioned in the introduction to this paper.

Suppose that the generalized diagram  $Vor(S)$  for a set  $S$  of circular discs has been constructed, and that it is represented by the data structures described in Section 4.

First consider the problem of detecting the existence of an intersection between any pair of discs in  $S$ . This can be tested using the following procedure:

(a) First check whether there exists a Voronoi cell  $V(i)$  having empty interior (i.e. a cell which is missing from  $Vor(S)$ ). If so,  $B_i$  is wholly contained in some other disc, and an intersection has been found. Otherwise, for each edge  $e$  in  $Vor(S)$  belonging to the common boundary of two Voronoi cells  $V(i)$  and  $V(j)$ , compute the value

$$\rho(e) = \min\{d(z, y) - r_i : y \in e\} = \min\{d(z, y) - r_j : y \in e\}.$$

If  $\rho(e) \leq 0$  for some  $e \in Vor(S)$ , then it is clear that the discs in  $S$  intersect. On the other hand, if  $\rho(e) > 0$  for each  $e \in Vor(S)$ , then no two discs in  $S$  intersect. Indeed, suppose that two discs  $B_i$  and  $B_j$  intersect each other. Let  $I$  be the segment  $[z_i, z_j]$ . For each  $y \in I$  consider the function

$$f(y) = \min_k \{d(z_k, y) - r_k : k = 1, 2, \dots, n\}.$$

Note that  $f(y) \leq 0$  for each  $y \in I$ , because each such  $y$  lies either in  $B_i$  or in  $B_j$ , so that either  $d(z_i, y) - r_i$  or  $d(z_j, y) - r_j$  is  $\leq 0$ . Moreover, by assumption both  $V(i)$  and  $V(j)$  have nonempty interiors, which implies by (2) that  $z_i$  belongs to  $V(i)$  and to no other cell. Hence  $I$  must intersect  $Vor(S)$  at least once. Let  $y \in I$  be a point belonging to some edge  $e$  of  $Vor(S)$ , and let  $V(i)$ ,  $V(j)$  be the two Voronoi cells containing  $y$  in their boundary. Then we have

$$\rho(e) \leq d(z_i, y) - r_i = f(y) \leq 0,$$

from which our claim follows immediately.

It is easy to see that a simple modification of the procedure just outlined yields a solution to the more complicated problem in which the discs in  $S$  are of several colors and we want to detect intersection between two discs of different colors. The appropriate procedure in this case is

(a) First check whether there exists a disc  $B_i$  which is wholly contained in another disc  $B_j$  of a different color, i.e. if  $V(i)$  has empty interior, and  $z_i$  belongs to a cell  $V(j)$  of a disc with a different color. Once the Voronoi diagram has been constructed by the method described in the preceding section, and has been supplemented by the mapping *contain*, we can detect such cases in  $O(n)$  time. Note that not every containment of a disc  $B_i$  of, say, red color in another disc  $B_j$  of a different color can be detected from the *contain* map, because  $B_i$  might be contained in another red disc  $B_k$ , and *contain* may map  $B_i$  directly to  $B_j$ . Nevertheless, if  $B_i$  is the *largest* possible disc contained in a disc of a different color, then *contain* will map  $B_i$  to some *differently-colored* disc containing it, so that if any disc is wholly contained in a disc of a different color, the procedure just described will detect at least one such situation.

(b) If step (a) detects no intersection, compute the quantities  $\rho(e)$ , as defined above, for all edges  $e \in Vor(S)$ . Then two discs of different colors intersect each other if and only if there exists an edge  $e$  common to two cells  $V(i)$  and  $V(j)$ , for which  $B_i$  and  $B_j$  have distinct colors, such that  $\rho(e) \leq 0$ .

**Proof:** Plainly if  $\rho(e) \leq 0$  for such an edge  $e$ , then  $e$  must contain a point  $y$  which lies inside both  $B_i$  and  $B_j$ , so that these two differently-colored discs intersect each other. Conversely, suppose that two differently-colored discs  $B_i$  and  $B_j$  intersect each other. Define the segment  $I$  and the function  $f$  on it as in the preceding paragraphs. We can assume without loss of generality that  $V(i)$  and  $V(j)$  have nonempty interiors, for if  $B_i$  (or  $B_j$ ) had empty interior then it would have been wholly contained in another disc  $B_k$  of the same color, so that we could replace  $B_i$  (or

$B_j$ ) by  $B_k$  in what follows. Call the colors of  $B_i$ ,  $B_j$  'red' and 'green', and call a Voronoi cell  $V(p)$  a 'red' (resp. 'green') cell if  $B_p$  is colored red (resp. green). Then  $z_i$  lies in (the interior of) a red cell, whereas  $z_j$  does not. Hence  $I$  must intersect  $Vor(S)$  at an edge  $e$  which separates a red cell  $V(k)$  and a cell  $V(l)$  of a different color. Arguing as before, it follows that  $\rho(e) \leq 0$  for this edge. Q.E.D.

Next consider the problem of determining the shortest distance between any two discs in  $S$ . Suppose that the discs in  $S$  do not intersect each other (if they do, the above procedures will detect this fact, and the distance that we seek will be 0), and let  $B_i$ ,  $B_j$  be the two discs closest to each other among all pairs of discs in  $S$ . Let  $y$  be the point on the segment  $I = [z_i, z_j]$  equidistant from  $B_i$  and  $B_j$ . We claim that  $y \in Vor(S)$ . For otherwise, there would exist another disc  $B_k \in S$  such that  $d(z_k, y) - r_k < d(z_i, y) - r_i$ . But then, by the triangle inequality,

$$\begin{aligned} d(B_j, B_k) &= d(z_j, z_k) - r_j - r_k \leq d(z_j, y) - r_j + d(z_i, y) - r_k \\ &< d(z_j, y) - r_j + d(z_i, y) - r_i = d(z_i, z_j) - r_i - r_j = d(B_i, B_j), \end{aligned}$$

contrary to assumption. Thus  $y \in Vor(S)$ . Moreover, the function

$$f(z) = \min\{d(z, z_k) - r_k : k = 1, \dots, n\}$$

attains its minimum value on the whole Voronoi diagram  $Vor(S)$  at the point  $y$ . This follows since by the triangle inequality we have  $2f(z) \geq d(B_i, B_j)$  for each  $z \in Vor(S)$ , where  $V(k)$  and  $V(l)$  are the two Voronoi cells containing  $z$ . It follows by the definition of  $f$  and by the preceding definition of  $\rho(e)$  that  $f(y)$  is the smallest of the values  $\rho(e)$ , for edges  $e$  of  $Vor(S)$ . Taken together, these arguments show that the shortest distance between two discs in  $S$  is equal to

$$\min\{2\rho(e) : e \text{ an edge of } Vor(S)\},$$

and hence this distance can be found in time  $O(n \log^2 n)$ .

A similar technique can be used to find the nearest neighbor in  $S$  of each  $B_i \in S$ . Indeed, an easy generalization of the preceding argument implies that if  $B_i$  is the nearest neighbor of  $B_j$ , then  $V(i)$  and  $V(j)$  meet at a common Voronoi edge, and the shortest distance between  $B_i$  and any other disc in  $S$  is equal to

$$\min\{2\rho(e) : e \text{ a boundary edge of } V(i)\},$$

from which the nearest neighbor of  $B_i$  is easily found.

These arguments extend easily to the case in which each of the discs of  $S$  is assigned a certain color, and we want to find the shortest distance between any two differently-colored discs. For this, let  $B_i$  and  $B_j$  be two discs in  $S$  of different colors such that their distance is the smallest among all distances between two differently-colored discs in  $S$ . Let the colors of  $B_i, B_j$  be 'red' and 'green' respectively. Let  $y$  be the point on  $[z_i, z_j]$  equidistant from  $B_i$  and  $B_j$ . We claim that  $y \in Vor(S)$ , for otherwise there would exist another disc  $B_k$  such that  $d(z_k, y) - r_k < d(z_i, y) - r_i = d(z_j, y) - r_j$ . If  $B_k$  is colored red, then arguing as above we would obtain  $d(B_j, B_k) < d(B_i, B_j)$ , i.e. a shorter distance between a red and a green disc. Similarly, if  $B_k$  is colored green, then we would have  $d(B_i, B_k) < d(B_i, B_j)$ , again a contradiction. Finally, if  $B_k$  is of another color, then both  $d(B_i, B_k)$  and  $d(B_j, B_k)$  are smaller than  $d(B_i, B_j)$ , again a contradiction.

Thus  $y \in Vor(S)$ , and similar arguments to those used above imply that  $f(y)$  is the minimum of all  $\rho(e)$ , for edges  $e$  of  $Vor(S)$  separating cells of different colors. This shows that the shortest distance between two differently-colored discs in  $S$  is simply the smallest of the values  $2\rho(e)$ , taken over all edges  $e$  of  $Vor(S)$  separating two differently-colored cells.

Other properties of generalized Voronoi diagrams deserve study. In particular, we would like to use these methods for solving the second part of Problem III mentioned in the introduction (that is, to find the differently-colored disc nearest to any given disc in  $S$ ), and for solving Problem IV, which calls for an appropriate generalization of known point-location algorithms.

## FOURTEEN DAYS

A fine will be charged for each day the book is kept overtime.

DEC 24 1982

Another interesting generalization of Voronoi diagrams along specified lines would facilitate collision-free motion of a robot system in a plane. The current algorithm, described in this section, is for the case in which the obstacles are sufficiently close to each other, along the specified lines.

It would also be interesting to consider the case of a set  $S$  of objects other than circles, such as "cigar-shaped" objects. Such generalizations of the algorithm, other than the current one, then the correctness of the algorithm for the case of  $n$  straight line segments is a remarkable property of the algorithm, even though they can be generalized to the case of arbitrary convex bodies.

### Acknowledgement

The author wishes to thank the referees for their careful reading of the manuscript and for their many helpful suggestions.

### References

- [Sh] M.I. Shamos, *Computational Geometry*, Wiley, New York, 1975.
- [Ki] D. Kirkpatrick, Efficient Computation of Continuous Skeletons, *Proc. 20th Symp. on Foundations of Computer Science* (1979), pp. 18-27.
- [NP] J. Nievergelt and F.P. Preparata, Plane-sweeping Algorithms for Intersecting Geometric Figures, *Comm. ACM* 25 (1982), pp. 739-747.
- [BO] J.L. Bentley and T.A. Ottman, Algorithms for Reporting and Counting Geometric Intersections, *IEEE Trans. Computers*, C-28 (1979), pp. 643-647.
- [LD] D.T. Lee and R.L. Drysdale III, Generalizations of Voronoi Diagrams in the Plane, *SIAM J. Computing*, 10 (1981), pp. 73-87.
- [Pr] F.P. Preparata, A New Approach to Planar Point Location, *SIAM J. Computing*, 10 (1981), pp. 473-482.
- [SS] J.T. Schwartz and M. Sharir, On the Piano Movers' Problem: I. The Special Case of a Rigid Polygonal Body Moving Amidst Polygonal Barriers, to appear in *Comm. Pure Appl. Math.*

the maintenance of the generalization of this latter problem to the case of a robot system in a plane. The current algorithm, described in this section, is for the case in which the obstacles are sufficiently close to each other, along the specified lines.

this paper is to the case of a set  $S$  of objects other than circles, such as "cigar-shaped" objects. Such generalizations of the algorithm, other than the current one, then the correctness of the algorithm for the case of  $n$  straight line segments is a remarkable property of the algorithm, even though they can be generalized to the case of arbitrary convex bodies.

the problems discussed in this paper, and for their many helpful suggestions.

GAYLORD 142

PRINTED IN U.S.A.

NYU  
Comp. Sci. Dept.  
TR-56 Sharir  
Intersection and closest-  
pair problems for a set...

c.1

c.1

NYU	
Comp. Sci. Dept.	
TR-56 Sharir	
AUTHOR	
Intersection and closest-	
TITLE	
pair problems for a set...	
DATE DUE	BORROWER'S NAME
12-2-81	Goodstadt

**N.Y.U. Courant Institute of  
Mathematical Sciences**  
251 Mercer St.  
New York, N. Y. 10012

2010-11-11